SPYWOLF

Security Audit Report

CONTRACT NOT DEPLOYED

APY

Completed on

June 29, 2022

MADE IN USA 🇺🇸

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

OVERVIEW

This audit has been prepared for **ApyMoon** to review the main aspects of the project to help investors make make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✓ Contract's source code
- ✓ Owners' wallets
- ✓ Tokenomics
- ✓ Team transparency and goals
- ✓ Website's age, code, security and UX
- ✓ Whitepaper and roadmap
- ✓ Social media & online presence

> *The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal*
>
> - SPYWOLF Team -

TABLE OF CONTENTS

ApyMoon

PROJECT DESCRIPTION

According to their whitepaper:
ApyMoon will be Yield Grow, Auto Staking, Anti Whale, Anti Dump, Auto Compounding Passive income generator.
Simple Moon Tokenomics for everyone, Buy $APYMOON Hold and watch your wallet Grow.
All Investors will be able to sell a MAX of 5%-20% of their current holdings over predefined period of time by the project's owners.

Release Date: Presale starts on July, 2022

Category: Rebase / Auto Staking

CONTRACT INFO

Token Name APYMOON	Symbol APYMOON
Contract Address NOT DEPLOYED YET	
Network NOT DEPLOYED YET	Language Solidity
Deployment Date NOT DEPLOYED YET	Verified? Yes
Total Supply 1,000,000	Status Not launched

TAXES

Buy Tax
12%

Sell Tax
13%

*Taxes can be changed in future

Our Contract Review Process

The contract review process pays special attention to the following:

- ✓ Testing the smart contracts against both common and uncommon vulnerabilities
- ✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
- ✓ Ensuring contract logic meets the specifications and intentions of the client.
- ✓ Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- ✓ Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

CURRENT STATS

(As of June 29, 2022)

Liquidity

Not added yet

Burn

No burnt tokens

Status:

Not Launched!

MaxSellTxAmount
1,000,000

DEX:
PancakeSwap

LP Address(es)

Liquidity not added yet

TOKEN TRANSFERS STATS

Transfer Count	Contract not deployed yet
Uniq Senders	Contract not deployed yet
Uniq Receivers	Contract not deployed yet
Total Amount	Contract not deployed yet
Median Transfer Amount	Contract not deployed yet
Average Transfer Amount	Contract not deployed yet
First transfer date	Contract not deployed yet
Last transfer date	Contract not deployed yet
Days token transferred	Contract not deployed yet

SMART CONTRACT STATS

Calls Count	Contract not deployed yet
External calls	Contract not deployed yet
Internal calls	Contract not deployed yet
Transactions count	Contract not deployed yet
Uniq Callers	Contract not deployed yet
Days contract called	Contract not deployed yet
Last transaction time	Contract not deployed yet
Created	Contract not deployed yet
Create TX	Contract not deployed yet
Creator	Contract not deployed yet

FEATURED WALLETS

Owner address	Assigned at deployment
Dev fee receiver	0x77fCfB635a43d1FbcEd5961754d4102C7e714b02
LP address	**Liquidity not added yet**

TOP 3 UNLOCKED WALLETS

1	N/A	N/A
2	N/A	N/A
3	N/A	N/A

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

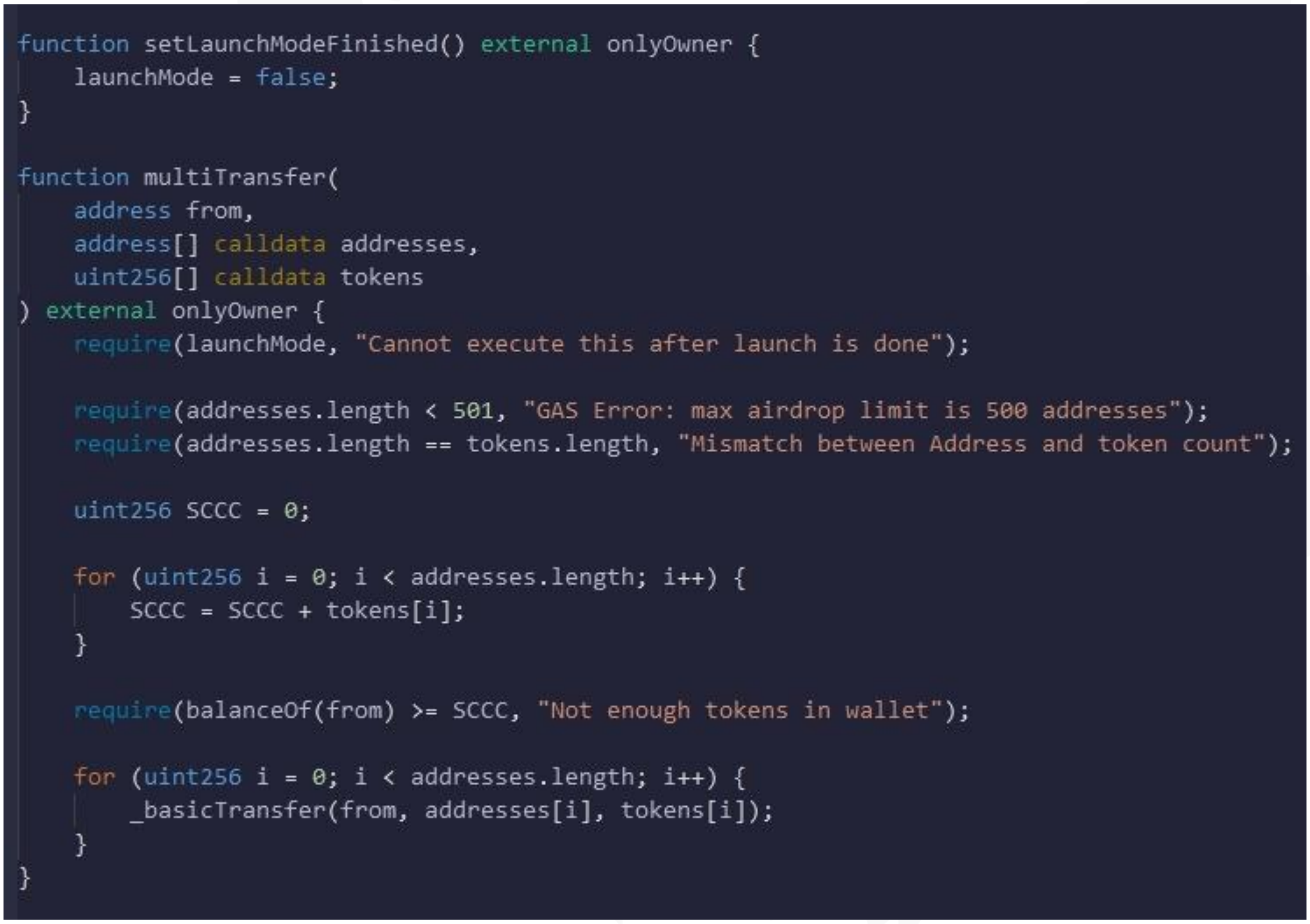
Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ High Risk

Owner can withdraw tokens from any address, including liquidity pair address and locking contracts - until the launchMode variable is set to false via setLaunchModeFinished() function.
Once the launchMode variable is set to false, owner cannot withdraw tokens from any address anymore.
The launchMode variable have no effect over actual token launch/trading.

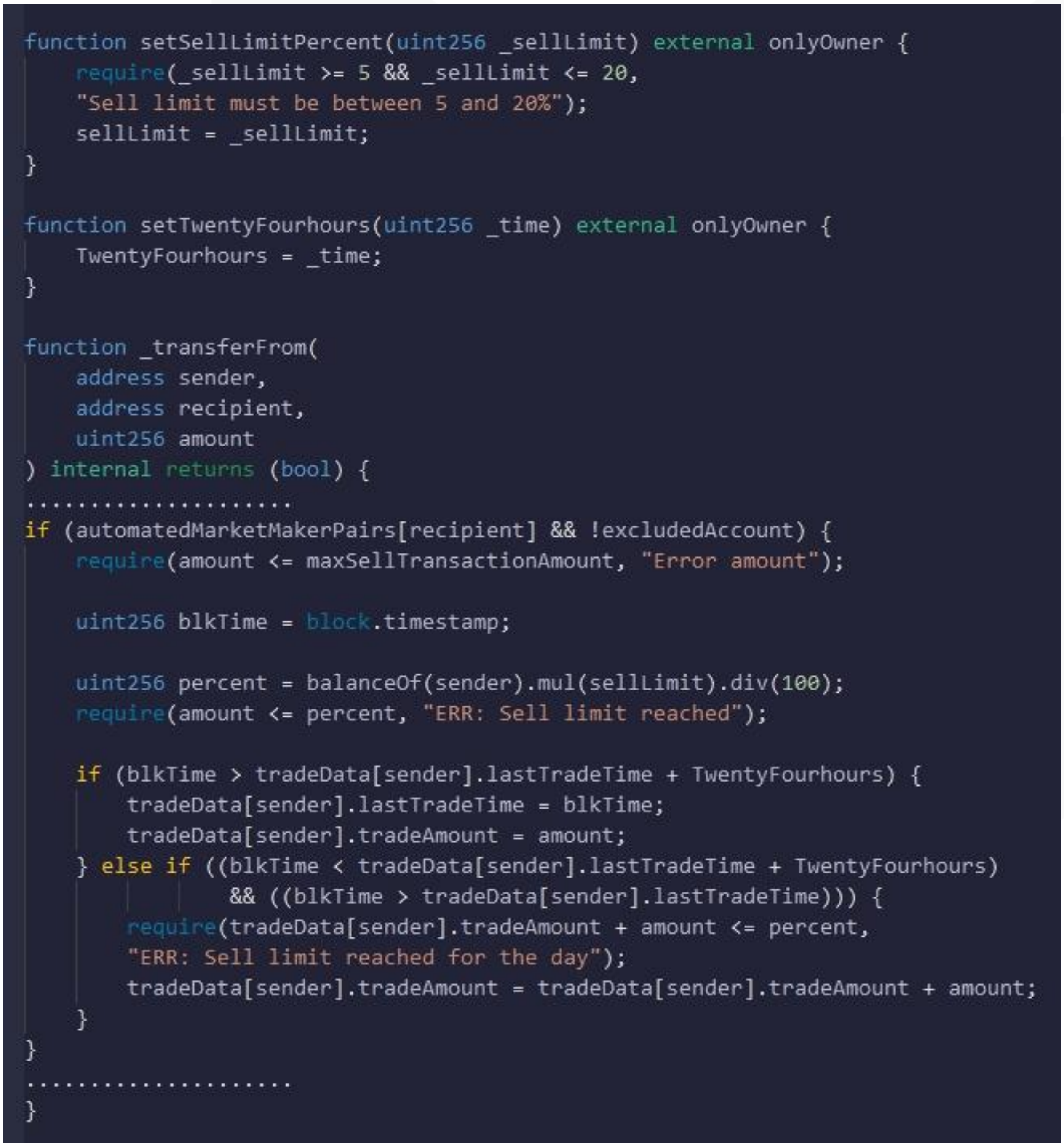


```
function setLaunchModeFinished() external onlyOwner {
    launchMode = false;
}

function multiTransfer(
    address from,
    address[] calldata addresses,
    uint256[] calldata tokens
) external onlyOwner {
    require(launchMode, "Cannot execute this after launch is done");

    require(addresses.length < 501, "GAS Error: max airdrop limit is 500 addresses");
    require(addresses.length == tokens.length, "Mismatch between Address and token count");

    uint256 SCCC = 0;

    for (uint256 i = 0; i < addresses.length; i++) {
        SCCC = SCCC + tokens[i];
    }

    require(balanceOf(from) >= SCCC, "Not enough tokens in wallet");

    for (uint256 i = 0; i < addresses.length; i++) {
        _basicTransfer(from, addresses[i], tokens[i]);
    }
}
```

FOUND THREATS

⚠ High Risk

Owner can change maximum sell amount percent of holder's holdings for period of time.
Sell amount can be between 5% to 20% of holder's balance.
Period of time can be up to 2^255 seconds (more than a year).

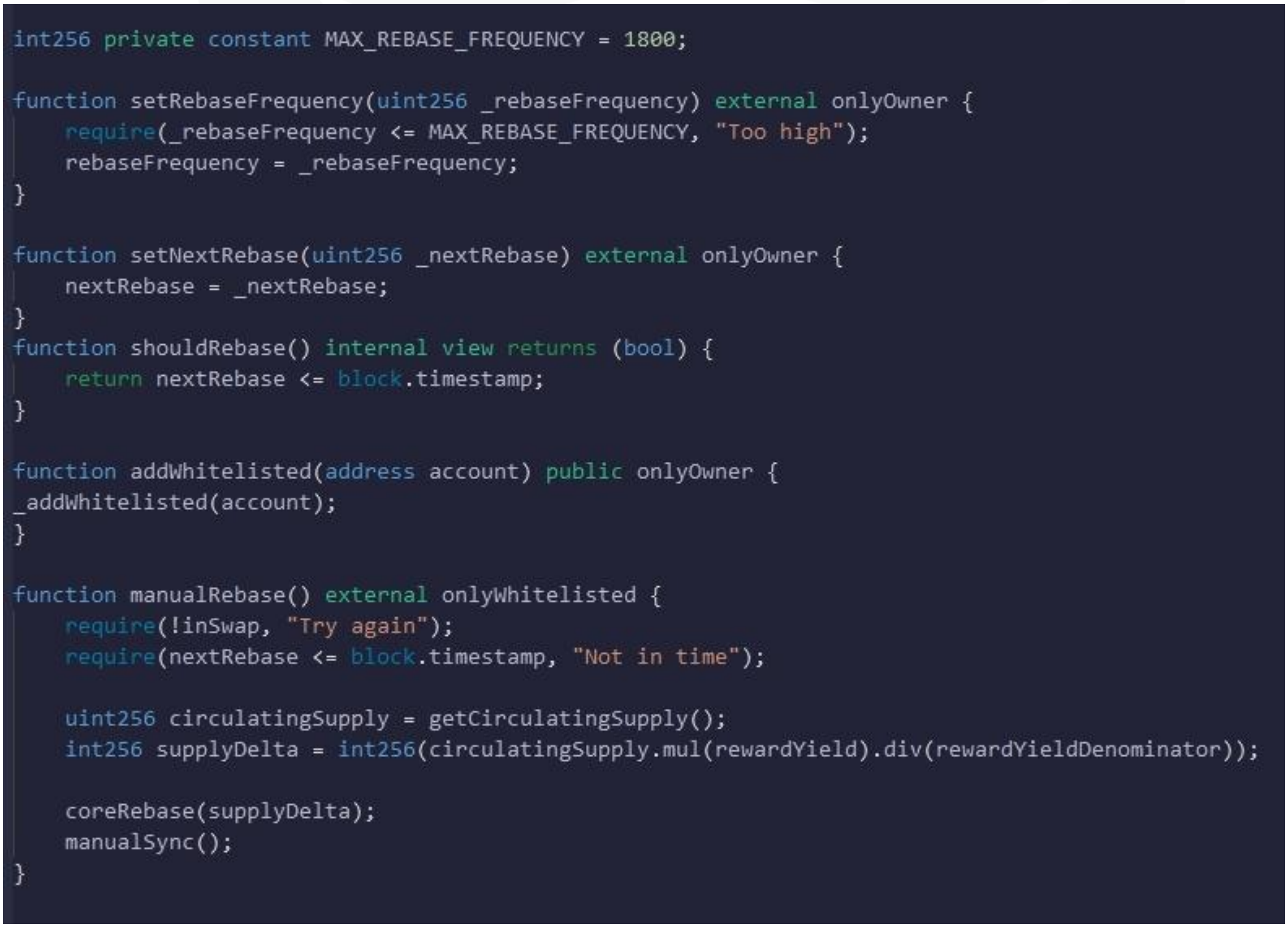


```
function setSellLimitPercent(uint256 _sellLimit) external onlyOwner {
    require(_sellLimit >= 5 && _sellLimit <= 20,
    "Sell limit must be between 5 and 20%");
    sellLimit = _sellLimit;
}

function setTwentyFourhours(uint256 _time) external onlyOwner {
    TwentyFourhours = _time;
}

function _transferFrom(
    address sender,
    address recipient,
    uint256 amount
) internal returns (bool) {
.....................
if (automatedMarketMakerPairs[recipient] && !excludedAccount) {
    require(amount <= maxSellTransactionAmount, "Error amount");

    uint256 blkTime = block.timestamp;

    uint256 percent = balanceOf(sender).mul(sellLimit).div(100);
    require(amount <= percent, "ERR: Sell limit reached");

    if (blkTime > tradeData[sender].lastTradeTime + TwentyFourhours) {
        tradeData[sender].lastTradeTime = blkTime;
        tradeData[sender].tradeAmount = amount;
    } else if ((blkTime < tradeData[sender].lastTradeTime + TwentyFourhours)
                && ((blkTime > tradeData[sender].lastTradeTime))) {
        require(tradeData[sender].tradeAmount + amount <= percent,
        "ERR: Sell limit reached for the day");
        tradeData[sender].tradeAmount = tradeData[sender].tradeAmount + amount;
    }
}
.....................
}
```

FOUND THREATS

⚠ Medium Risk

Owner can change rebase settings:
Owner can change rebase frequency from 0 to 1800 seconds.
Owner can set time for rebase.
Owner can assign whitelisted address.
Whitelisted address can initiate manual rebase.

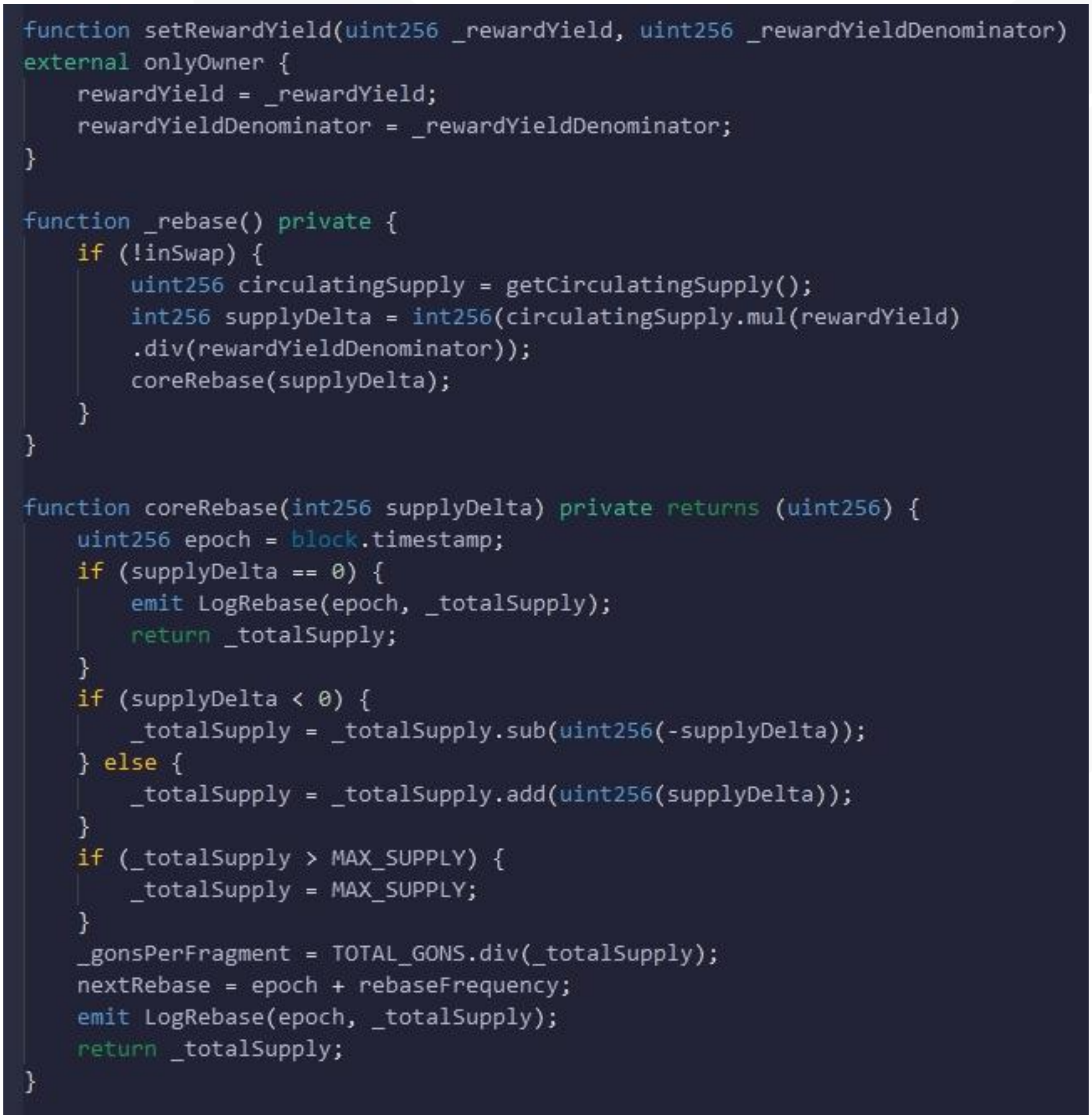


```
int256 private constant MAX_REBASE_FREQUENCY = 1800;

function setRebaseFrequency(uint256 _rebaseFrequency) external onlyOwner {
    require(_rebaseFrequency <= MAX_REBASE_FREQUENCY, "Too high");
    rebaseFrequency = _rebaseFrequency;
}

function setNextRebase(uint256 _nextRebase) external onlyOwner {
    nextRebase = _nextRebase;
}
function shouldRebase() internal view returns (bool) {
    return nextRebase <= block.timestamp;
}

function addWhitelisted(address account) public onlyOwner {
_addWhitelisted(account);
}

function manualRebase() external onlyWhitelisted {
    require(!inSwap, "Try again");
    require(nextRebase <= block.timestamp, "Not in time");

    uint256 circulatingSupply = getCirculatingSupply();
    int256 supplyDelta = int256(circulatingSupply.mul(rewardYield).div(rewardYieldDenominator));

    coreRebase(supplyDelta);
    manualSync();
}
```

FOUND THREATS

⚠ Medium Risk

Owner can change rebase settings:
Owner can change token's rebase rate - how many tokens are added to the total supply on each rebase.

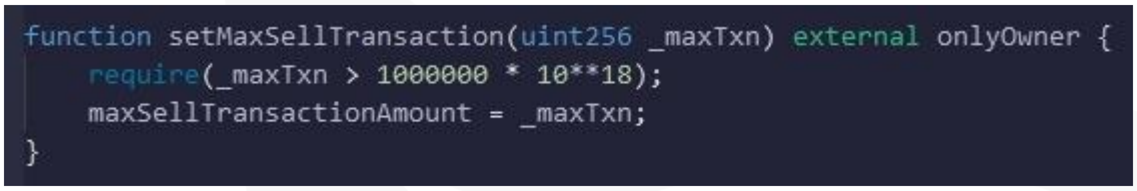


```
function setRewardYield(uint256 _rewardYield, uint256 _rewardYieldDenominator)
external onlyOwner {
    rewardYield = _rewardYield;
    rewardYieldDenominator = _rewardYieldDenominator;
}

function _rebase() private {
    if (!inSwap) {
        uint256 circulatingSupply = getCirculatingSupply();
        int256 supplyDelta = int256(circulatingSupply.mul(rewardYield)
        .div(rewardYieldDenominator));
        coreRebase(supplyDelta);
    }
}

function coreRebase(int256 supplyDelta) private returns (uint256) {
    uint256 epoch = block.timestamp;
    if (supplyDelta == 0) {
        emit LogRebase(epoch, _totalSupply);
        return _totalSupply;
    }
    if (supplyDelta < 0) {
        _totalSupply = _totalSupply.sub(uint256(-supplyDelta));
    } else {
        _totalSupply = _totalSupply.add(uint256(supplyDelta));
    }
    if (_totalSupply > MAX_SUPPLY) {
        _totalSupply = MAX_SUPPLY;
    }
    _gonsPerFragment = TOTAL_GONS.div(_totalSupply);
    nextRebase = epoch + rebaseFrequency;
    emit LogRebase(epoch, _totalSupply);
    return _totalSupply;
}
```

FOUND THREATS

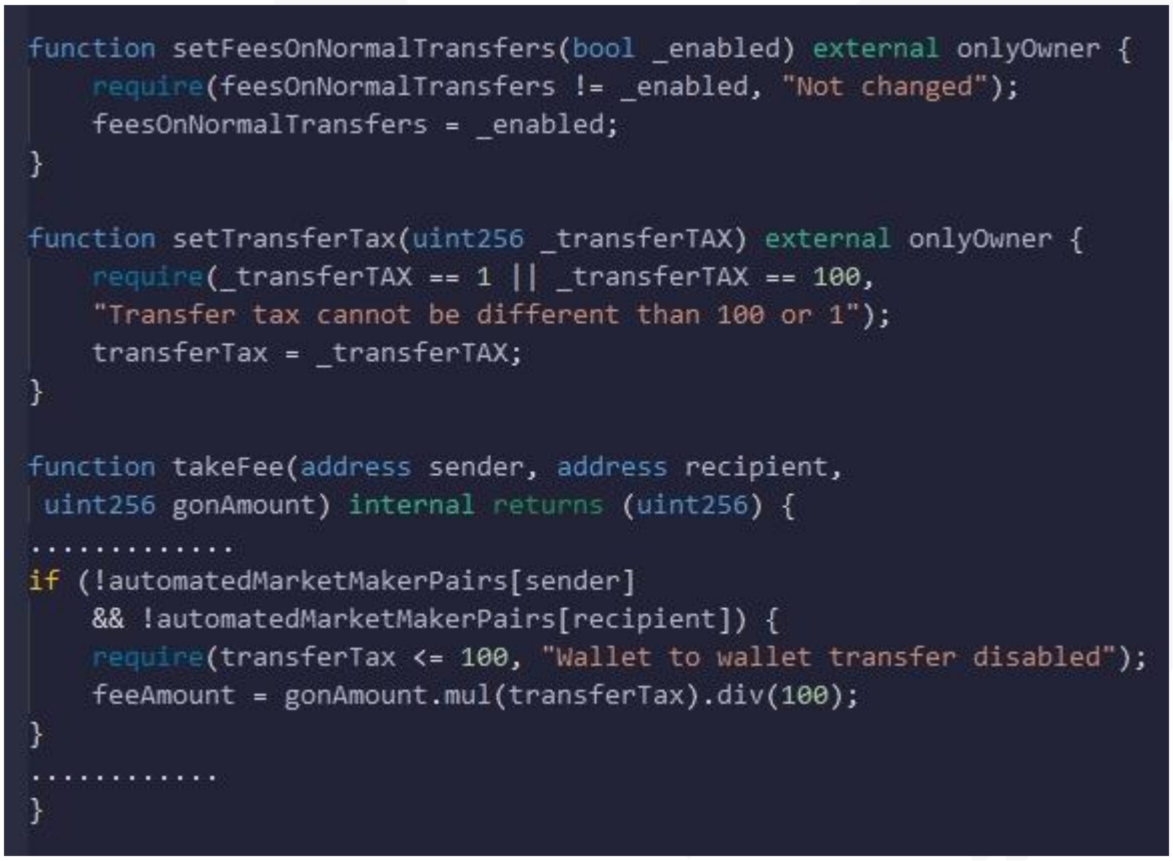
⚠ Medium Risk

Owner can change max transaction limit, but can't lower it than 1 million tokens.



```
function setMaxSellTransaction(uint256 _maxTxn) external onlyOwner {
    require(_maxTxn > 1000000 * 10**18);
    maxSellTransactionAmount = _maxTxn;
}
```

Owner can disable transfers between wallet to wallet.
This don't affect buys/sells.

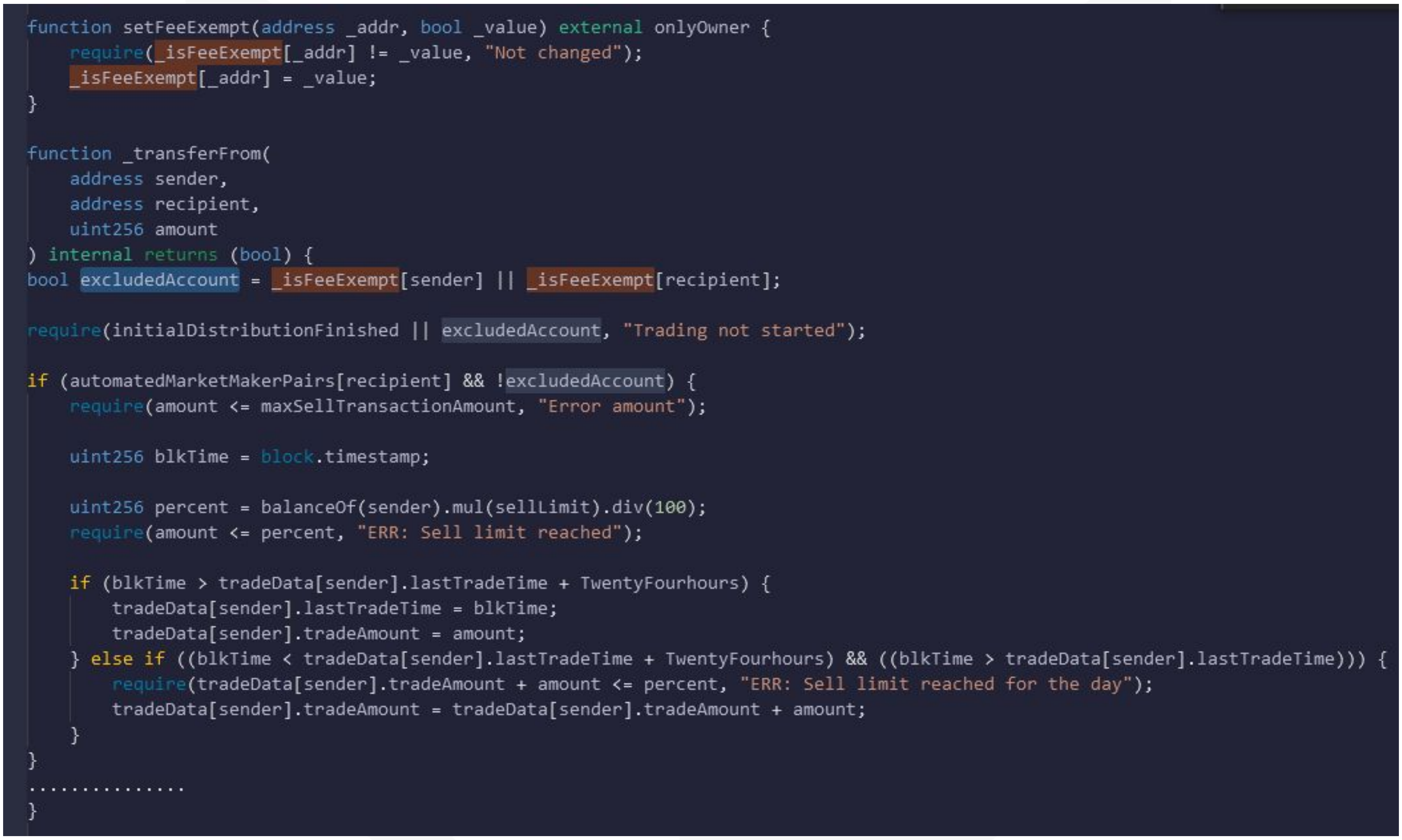


```
function setFeesOnNormalTransfers(bool _enabled) external onlyOwner {
    require(feesOnNormalTransfers != _enabled, "Not changed");
    feesOnNormalTransfers = _enabled;
}

function setTransferTax(uint256 _transferTAX) external onlyOwner {
    require(_transferTAX == 1 || _transferTAX == 100,
    "Transfer tax cannot be different than 100 or 1");
    transferTax = _transferTAX;
}

function takeFee(address sender, address recipient,
 uint256 gonAmount) internal returns (uint256) {
.............
if (!automatedMarketMakerPairs[sender]
    && !automatedMarketMakerPairs[recipient]) {
    require(transferTax <= 100, "Wallet to wallet transfer disabled");
    feeAmount = gonAmount.mul(transferTax).div(100);
}
...........
}
```

FOUND THREATS

Medium Risk

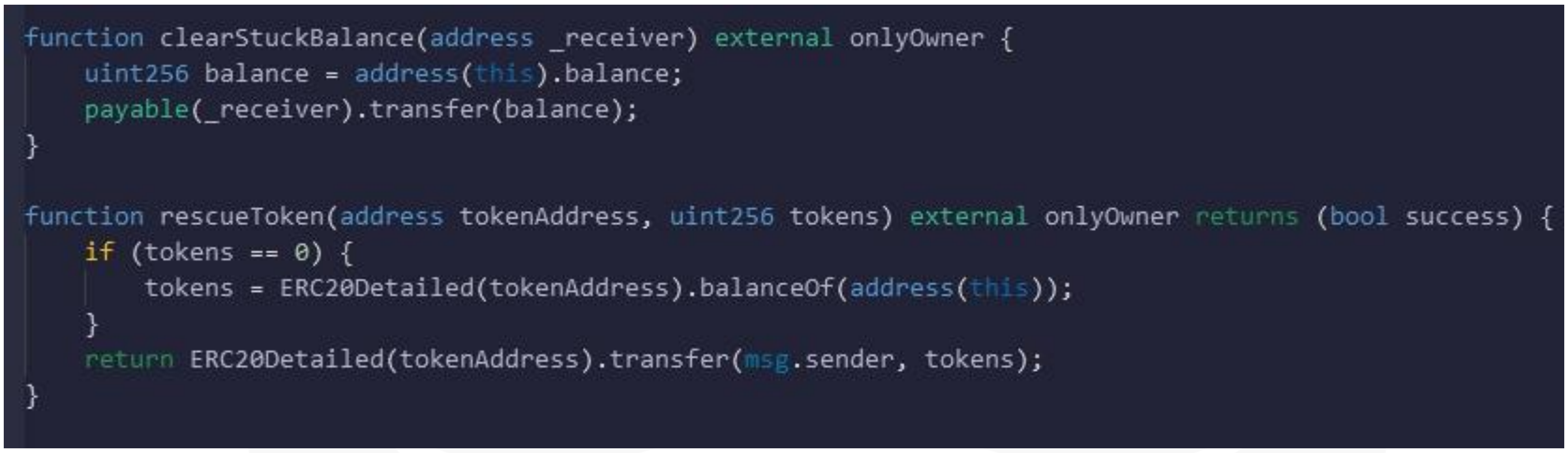
Owner can exclude wallets from taxes and sell limitations.



```
function setFeeExempt(address _addr, bool _value) external onlyOwner {
    require(_isFeeExempt[_addr] != _value, "Not changed");
    _isFeeExempt[_addr] = _value;
}

function _transferFrom(
    address sender,
    address recipient,
    uint256 amount
) internal returns (bool) {
bool excludedAccount = _isFeeExempt[sender] || _isFeeExempt[recipient];

require(initialDistributionFinished || excludedAccount, "Trading not started");

if (automatedMarketMakerPairs[recipient] && !excludedAccount) {
    require(amount <= maxSellTransactionAmount, "Error amount");

    uint256 blkTime = block.timestamp;

    uint256 percent = balanceOf(sender).mul(sellLimit).div(100);
    require(amount <= percent, "ERR: Sell limit reached");

    if (blkTime > tradeData[sender].lastTradeTime + TwentyFourhours) {
        tradeData[sender].lastTradeTime = blkTime;
        tradeData[sender].tradeAmount = amount;
    } else if ((blkTime < tradeData[sender].lastTradeTime + TwentyFourhours) && ((blkTime > tradeData[sender].lastTradeTime))) {
        require(tradeData[sender].tradeAmount + amount <= percent, "ERR: Sell limit reached for the day");
        tradeData[sender].tradeAmount = tradeData[sender].tradeAmount + amount;
    }
}
...............
}
```

Low Risk

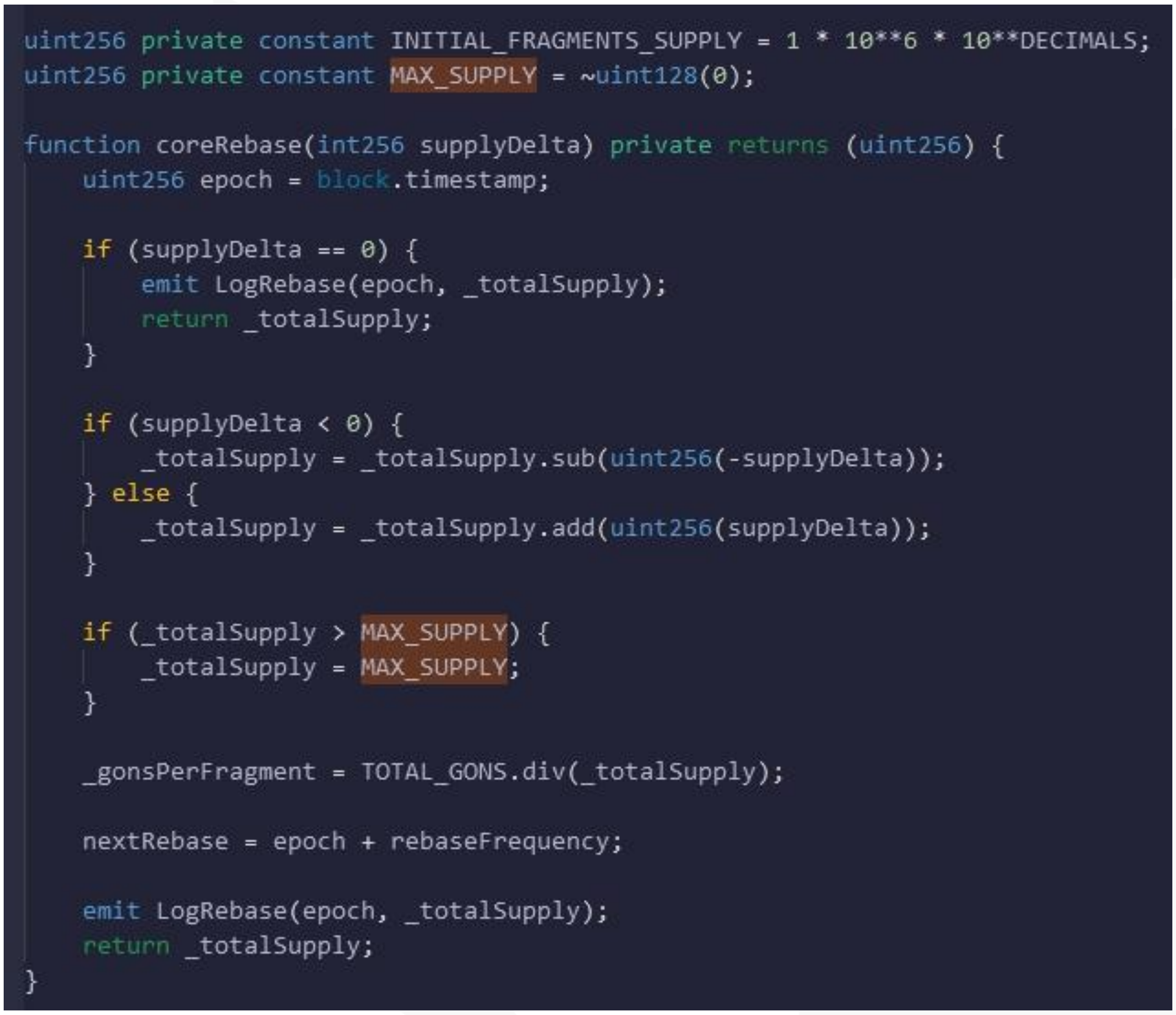
Owner can withdraw any tokens from the contract.



```
function clearStuckBalance(address _receiver) external onlyOwner {
    uint256 balance = address(this).balance;
    payable(_receiver).transfer(balance);
}

function rescueToken(address tokenAddress, uint256 tokens) external onlyOwner returns (bool success) {
    if (tokens == 0) {
        tokens = ERC20Detailed(tokenAddress).balanceOf(address(this));
    }
    return ERC20Detailed(tokenAddress).transfer(msg.sender, tokens);
}
```

Informational

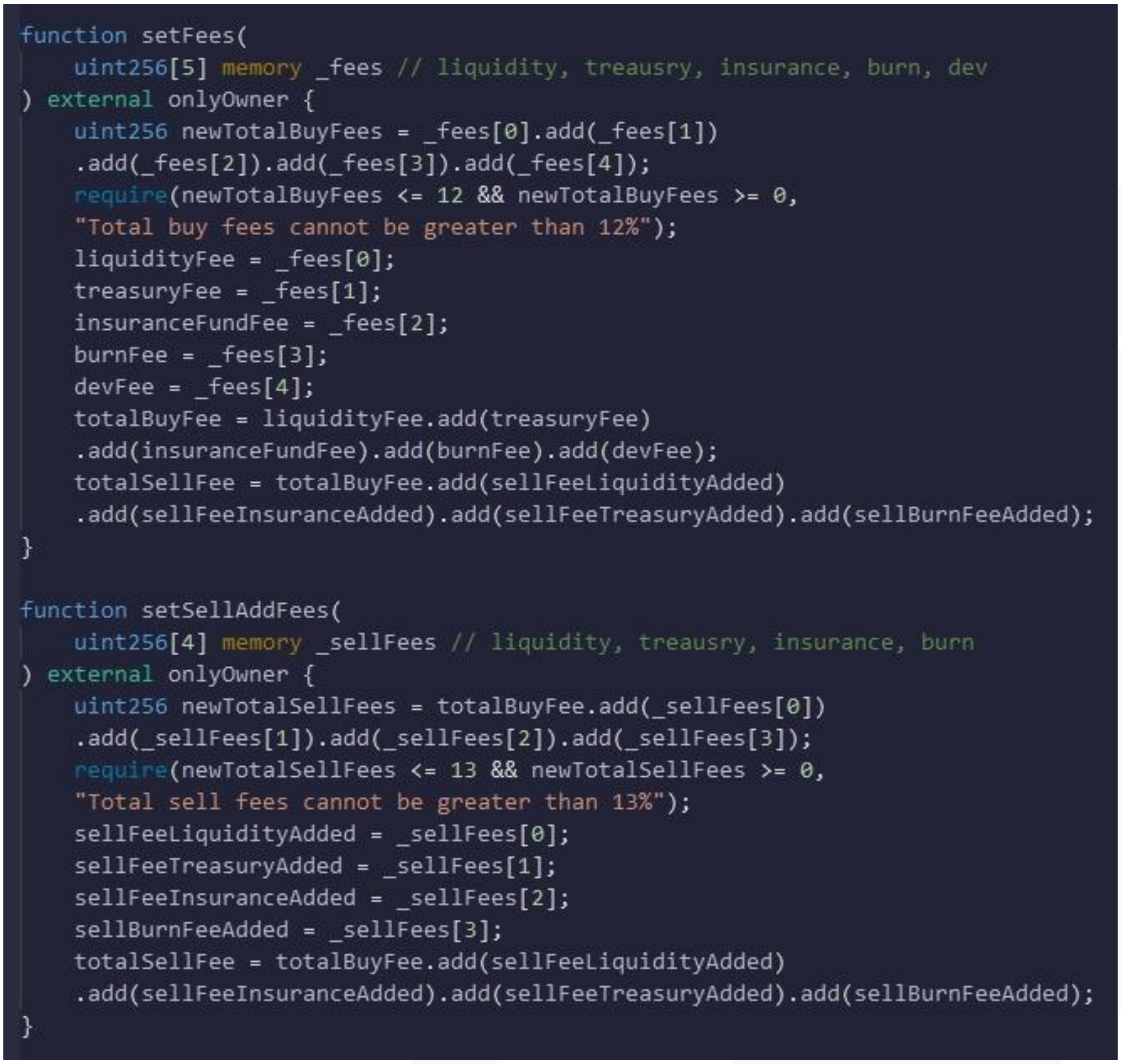
This is rebase token with MAX_SUPPLY up to 340282366920938463463374607431768211456.
Current supply is 1000000.
Rebase tokens can lead to price inflation in future.



```
uint256 private constant INITIAL_FRAGMENTS_SUPPLY = 1 * 10**6 * 10**DECIMALS;
uint256 private constant MAX_SUPPLY = ~uint128(0);

function coreRebase(int256 supplyDelta) private returns (uint256) {
    uint256 epoch = block.timestamp;

    if (supplyDelta == 0) {
        emit LogRebase(epoch, _totalSupply);
        return _totalSupply;
    }

    if (supplyDelta < 0) {
        _totalSupply = _totalSupply.sub(uint256(-supplyDelta));
    } else {
        _totalSupply = _totalSupply.add(uint256(supplyDelta));
    }

    if (_totalSupply > MAX_SUPPLY) {
        _totalSupply = MAX_SUPPLY;
    }

    _gonsPerFragment = TOTAL_GONS.div(_totalSupply);

    nextRebase = epoch + rebaseFrequency;

    emit LogRebase(epoch, _totalSupply);
    return _totalSupply;
}
```

Informational

Owner can set buy fees up to 12% and sell fees up to 13%.
Combined buy+sell=25%.



```
function setFees(
    uint256[5] memory _fees // liquidity, treausry, insurance, burn, dev
) external onlyOwner {
    uint256 newTotalBuyFees = _fees[0].add(_fees[1])
    .add(_fees[2]).add(_fees[3]).add(_fees[4]);
    require(newTotalBuyFees <= 12 && newTotalBuyFees >= 0,
    "Total buy fees cannot be greater than 12%");
    liquidityFee = _fees[0];
    treasuryFee = _fees[1];
    insuranceFundFee = _fees[2];
    burnFee = _fees[3];
    devFee = _fees[4];
    totalBuyFee = liquidityFee.add(treasuryFee)
    .add(insuranceFundFee).add(burnFee).add(devFee);
    totalSellFee = totalBuyFee.add(sellFeeLiquidityAdded)
    .add(sellFeeInsuranceAdded).add(sellFeeTreasuryAdded).add(sellBurnFeeAdded);
}

function setSellAddFees(
    uint256[4] memory _sellFees // liquidity, treausry, insurance, burn
) external onlyOwner {
    uint256 newTotalSellFees = totalBuyFee.add(_sellFees[0])
    .add(_sellFees[1]).add(_sellFees[2]).add(_sellFees[3]);
    require(newTotalSellFees <= 13 && newTotalSellFees >= 0,
    "Total sell fees cannot be greater than 13%");
    sellFeeLiquidityAdded = _sellFees[0];
    sellFeeTreasuryAdded = _sellFees[1];
    sellFeeInsuranceAdded = _sellFees[2];
    sellBurnFeeAdded = _sellFees[3];
    totalSellFee = totalBuyFee.add(sellFeeLiquidityAdded)
    .add(sellFeeInsuranceAdded).add(sellFeeTreasuryAdded).add(sellBurnFeeAdded);
}
```

RECOMMENDATIONS FOR

GOOD PRACTICES

1 Consider fundamental tradeoffs

2 Be attentive to blockchain properties

3 Ensure careful rollouts

4 Keep contracts simple

5 Stay up to date and track development

Apy Moon

GOOD PRACTICES FOUND

✓ The owner cannot stop or pause the contract

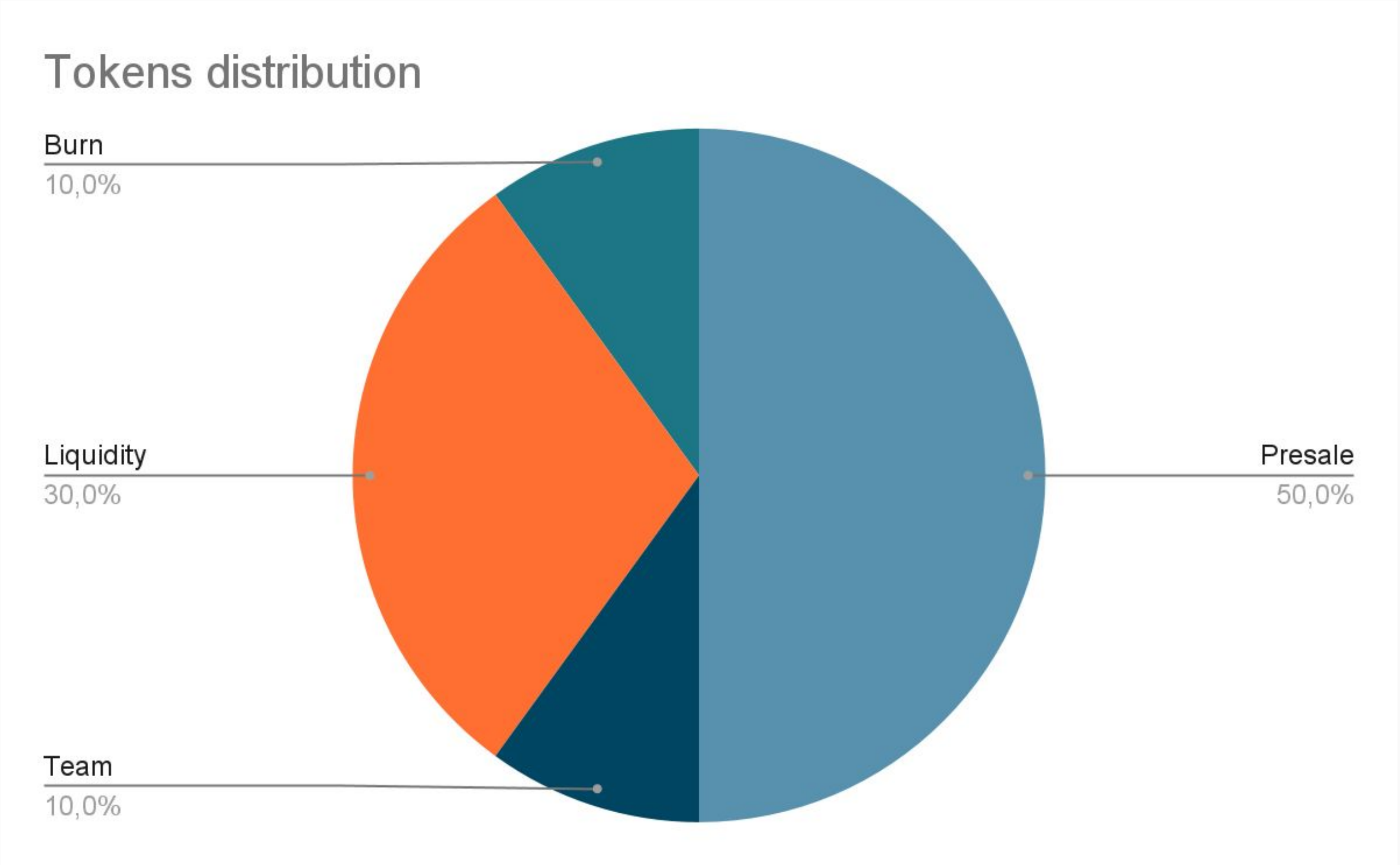
✓ The owner can set a transaction limit, but can’t lower it than 1 million tokens

✓ The smart contract utilizes "SafeMath" to prevent overflows

TOKENOMICS

*The following tokenomics are based on the project's whitepaper and/or website:

- 50% - Presale
- 30% -Liquidity
- 10% - Team
- 10% - Burn



THE TEAM

The team has privately doxxed to SPYWOLF by completing the following KYC requirements:

- ID Verification
- Video statement
- Video interview with devs
- Owner's wallet verification

KYC INFORMATION

Issuer

SPYWOLF

Members KYC'd

👤

KYC Date

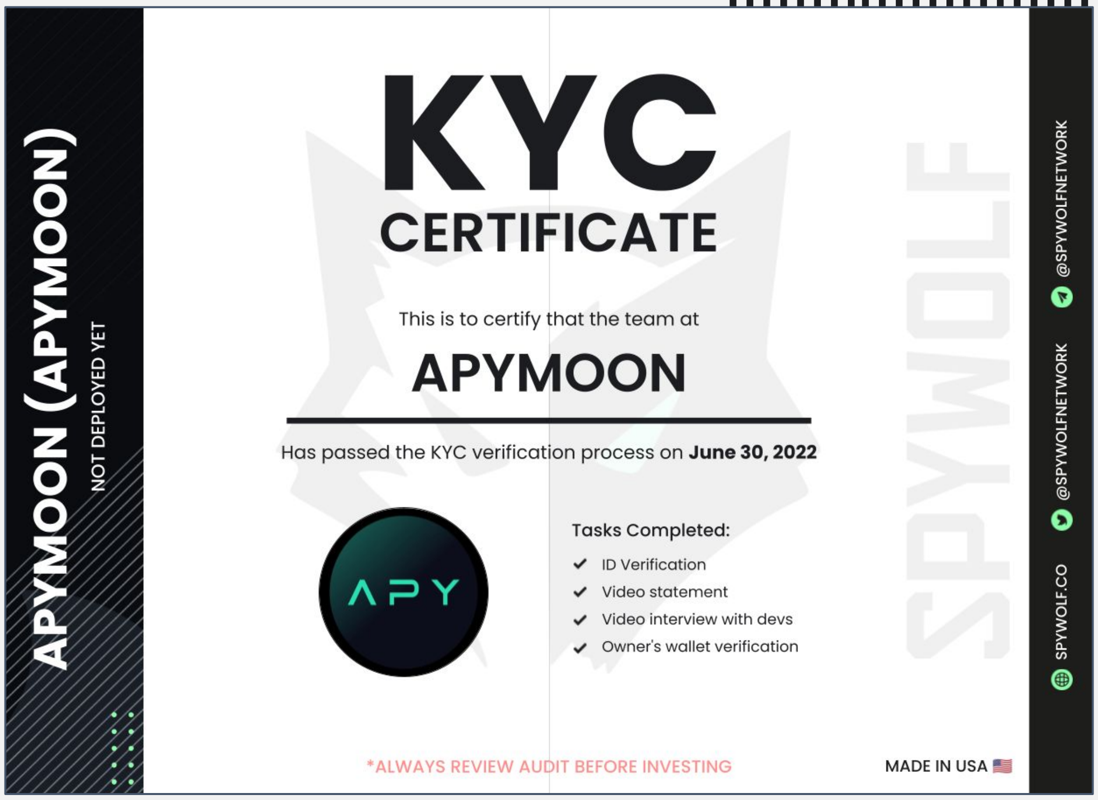
June 30, 2022

Format

Image

Certificate Link

https://github.com/SpyWolfNetwork/KYCs/blob/main/june/KYC_APYMOON_NOTDEPLOYEDYET.png



APYMOON (APYMOON)

NOT DEPLOYED YET

KYC

CERTIFICATE

This is to certify that the team at

APYMOON

Has passed the KYC verification process on **June 30, 2022**

Tasks Completed:

- ✓ ID Verification
- ✓ Video statement
- ✓ Video interview with devs
- ✓ Owner's wallet verification

*ALWAYS REVIEW AUDIT BEFORE INVESTING

MADE IN USA

SPYWOLF

@SPYWOLFNETWORK

@SPYWOLFNETWORK

SPYWOLF.CO

WEBSITE

Website URL
https://www.apymoon.com/

Domain Registry
https://www.namesilo.com

Domain Expiration
Expires on 2023-04-28

Technical SEO Test
Passed

Security Test
Passed. SSL certificate present

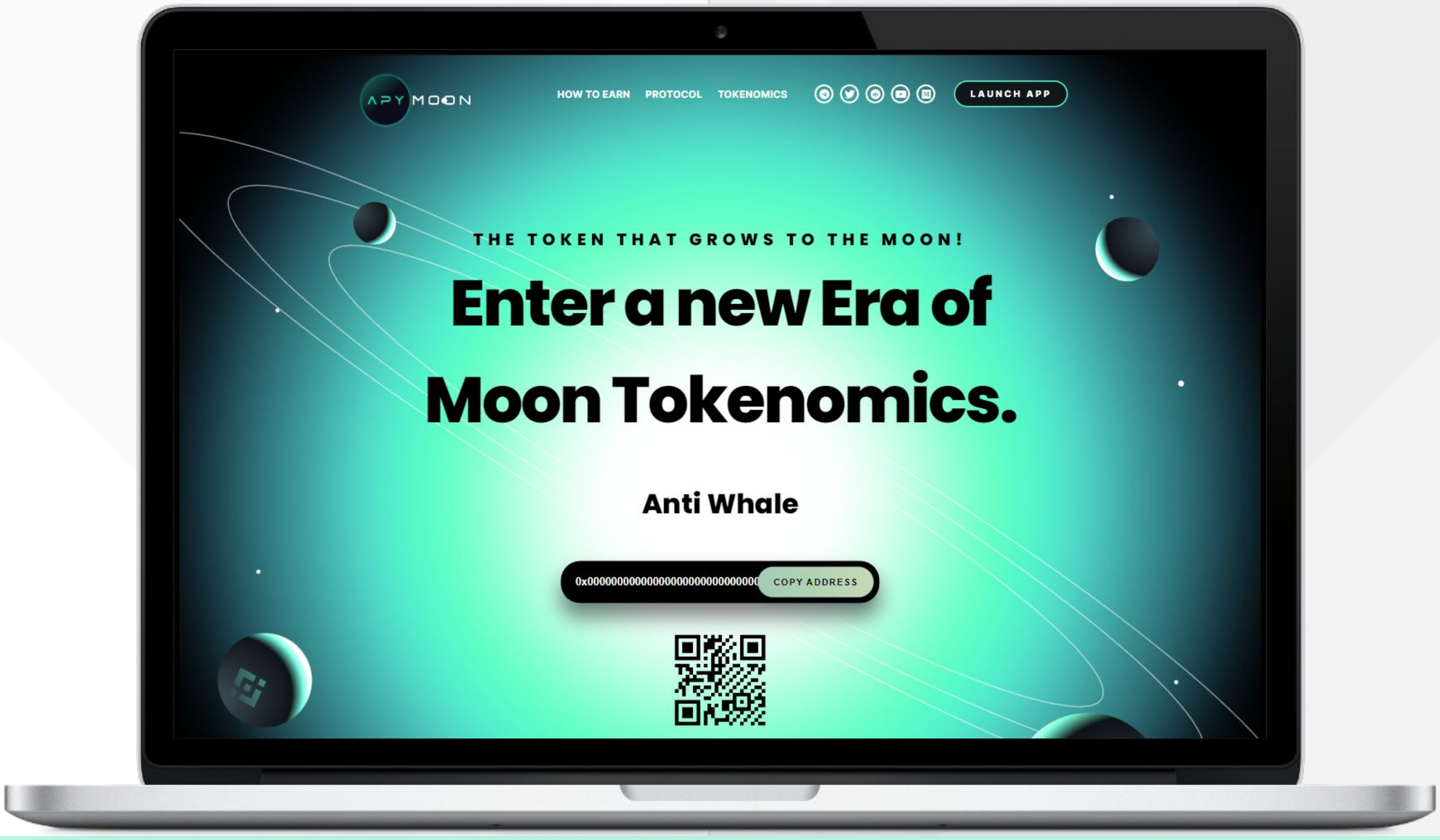
Design
Very nice color scheme and overall layout.

Content
The information helps new investors understand what the product does right away. No grammar mistakes found.

Whitepaper
Partial, not very explanatory, bit short.

Roadmap
Partial, goals set for the 1st phase.

Mobile-friendly?
Yes



THE TOKEN THAT GROWS TO THE MOON!

Enter a new Era of Moon Tokenomics.

Anti Whale

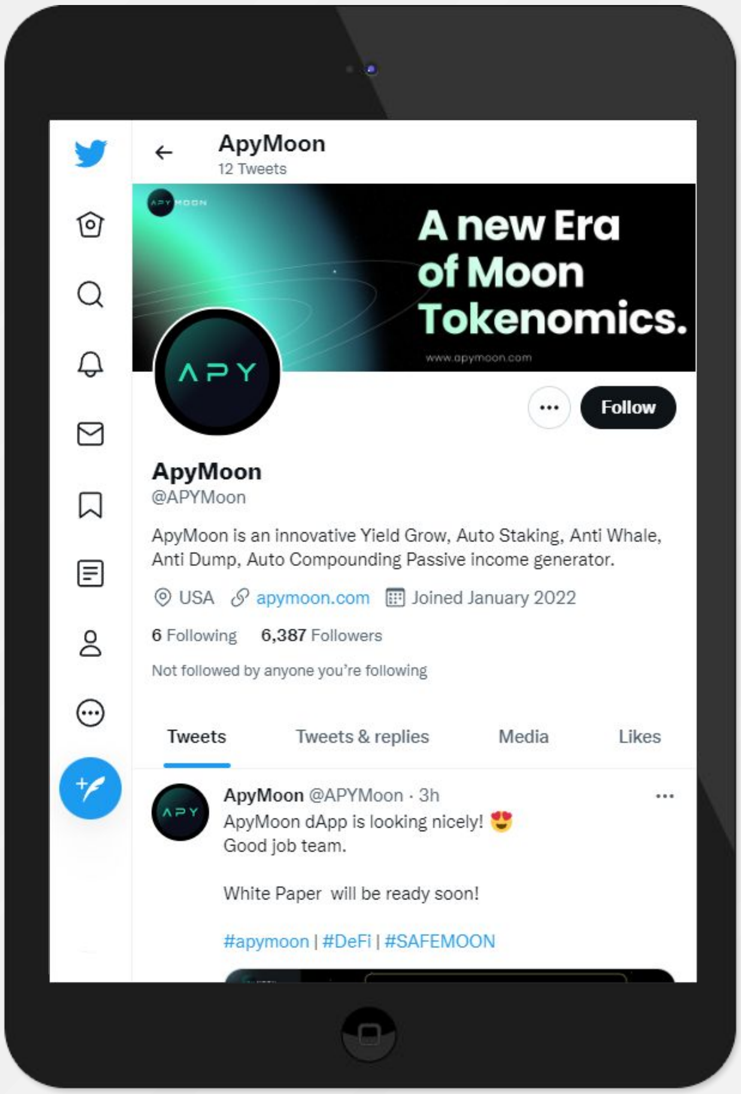
apymoon.com

SOCIAL MEDIA

& ONLINE PRESENCE

ANALYSIS

Project's social media presence is relatively new (few days old). The mods are active, but there are few organic interactions, and bot-like behaviors were detected ⚠️.



ApyMoon
12 Tweets

A new Era of Moon Tokenomics.
www.apymoon.com

ApyMoon
@APYMoon

ApyMoon is an innovative Yield Grow, Auto Staking, Anti Whale, Anti Dump, Auto Compounding Passive income generator.

USA apymoon.com Joined January 2022

6 Following 6,387 Followers

Not followed by anyone you're following

ApyMoon @APYMoon · 3h
ApyMoon dApp is looking nicely! 😍
Good job team.

White Paper will be ready soon!

#apymoon | #DeFi | #SAFEMOON

Twitter

@APYMoon

- 6,387 followers
- Recently active - 12 tweets, last one June 28
- Few active followers - doesn't correspond to followers number ⚠️

Discord

Discord link here

- Not available

Telegram

@ApyMoon

- 5 members
- Recently created, 0 messages
- No activity registered

Medium

@apymoon

- 0 followers
- Recently created, 4 posts

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- ✓ OVER 150 SUCCESSFUL CLIENTS
- ✓ MORE THAN 500 SCAMS EXPOSED
- ✓ MILLIONS SAVED IN POTENTIAL FRAUD
- ✓ PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS
- ✓ CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO	SPYWOLF.NETWORK
@SPYWOLFNETWORK	@SPYWOLFOFFICIAL
@SPYWOLFNETWORK	@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.

While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.

No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.

The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.

No applications were reviewed for security. No product code has been reviewed.